

TEMAS

Mongo



Express



Angular



Node



PHP



JavaScript



HTML5



CSS3



Sublime Text



Bootstrap



Grunt



Vagrant

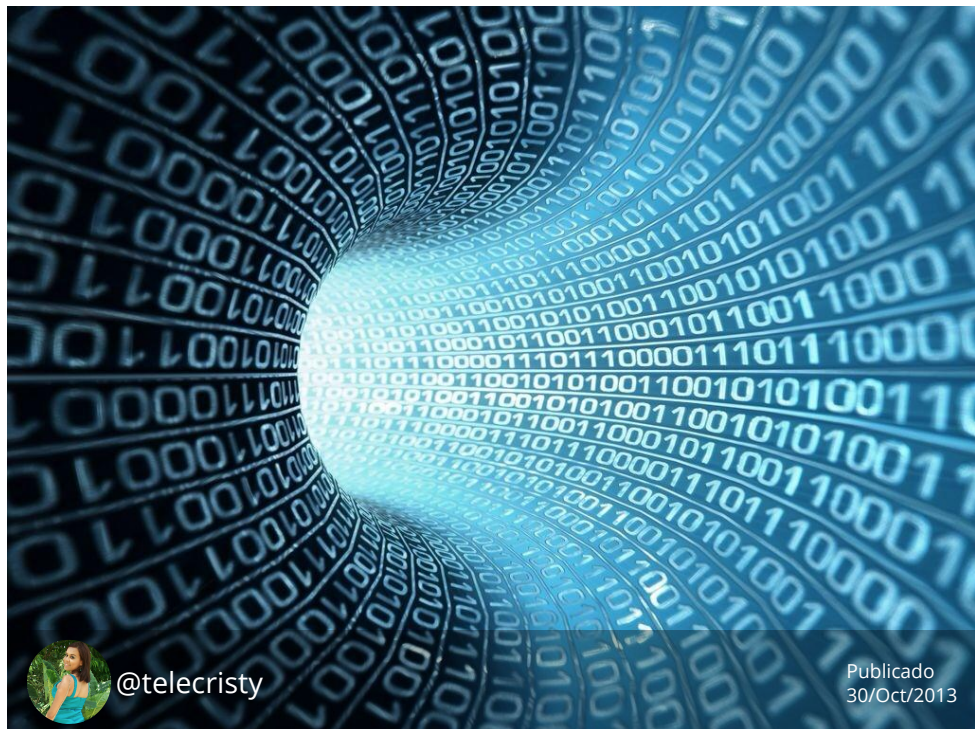


Docker



PATROCINADORES

Creación y recorrido de un árbol binario en C++



@telecristy

Publicado
30/Oct/2013

COMPARTIR EN FACEBOOK

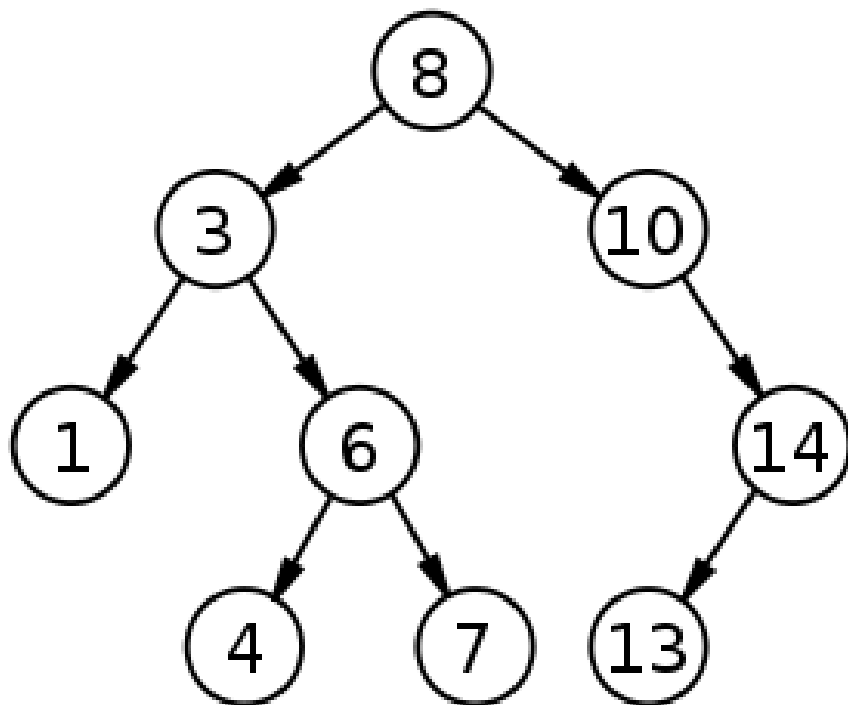


COMPARTIR EN TWITTER



COMPARTIR EN GOOGLE+

Esta publicación tiene 43 meses de antigüedad, te invitamos a buscar publicaciones más recientes relacionados a este tema [aquí](#).



Un árbol binario de búsqueda y lo recorremos de tres formas: inorder, en preorden y en postorden, un saludo de mi parte @telecristy.

```
/*Crea un arbol binario y lo recorre en  
preorden, inorden, y en postOrden */  
#include <stdio.h>
```



```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

/* estructura autoreferenciada */
struct nodoArbol {
    struct nodoArbol *ptrIzq; /* apuntador al subárbol izquierdo */
    int dato; /* valor del nodo */
    struct nodoArbol *ptrDer; /* apuntador al subárbol derecho */
}; /* fin de la estructura nodoArbol */

typedef struct nodoArbol NodoArbol; /* sinónimo de la estructura nodoArbol */
typedef NodoArbol *ptrNodoArbol; /* sinónimo de NodoArbol* */</time.h></stdlib.h></stdio.h>
```

```
/* prototipos */
void insertaNodo( ptrNodoArbol *ptrArbol, int valor );
void inOrden( ptrNodoArbol ptrArbol );
void preOrden( ptrNodoArbol ptrArbol );
void postOrden( ptrNodoArbol ptrArbol );

/* la función main comienza la ejecución del programa */
int main()
{
    int i; /* contador para el ciclo de 1 a 10 */
    int elemento; /* variable para almacenar valores al azar */
    ptrNodoArbol ptrRaiz = NULL; /* árbol inicialmente vacío */

    srand( time( NULL ) );
    printf( "Los números colocados en el árbol son:
" );

    /* inserta valores al azar entre 1 y 15 en el árbol */
    for ( i = 1; i <= 10; i++ ) {
        elemento = rand() % 15;
        printf( "%3d", elemento );
        insertaNodo( &ptrRaiz, elemento );
    } /* fin de for */

    /* recorre el árbol en preorden */
    printf( "

El recorrido en preorden es:
" );
    preOrden( ptrRaiz );

    /* recorre el árbol en in inorden */
    printf( "

El recorrido inorden es:
" );
    inOrden( ptrRaiz );

    /* recorre el árbol en postOrden */
    printf( "

El recorrido en postOrden es:
" );
    postOrden( ptrRaiz );

    return 0; /* indica terminación exitosa */
} /* fin de main */

/* inserta un nodo dentro del árbol */
void insertaNodo( ptrNodoArbol *ptrArbol, int valor )
{
    /* si el árbol está vacío */
    if ( *ptrArbol == NULL ) {
        *ptrArbol = malloc( sizeof( NodoArbol ) );

        /* si la memoria está asignada, entonces asigna el dato */
        if ( *ptrArbol != NULL ) {
            ( *ptrArbol )->dato = valor;
            ( *ptrArbol )->ptrIzq = NULL;
            ( *ptrArbol )->ptrDer = NULL;
        } /* fin de if */
        else {
            printf( "no se inserto %d. No hay memoria disponible.
", valor );
        } /* fin de else */

    } /* fin de if */
    else { /* el árbol no está vacío */

        /* el dato a insertar es menor que el dato en el nodo actual */
        if ( valor < ( *ptrArbol )->dato ) {
            insertaNodo( &( ( *ptrArbol )->ptrIzq ), valor );
        } /* fin de if */

        /* el dato a insertar es mayor que el dato en el nodo actual */
        else if ( valor > ( *ptrArbol )->dato ) {
            insertaNodo( &( ( *ptrArbol )->ptrDer ), valor );
        } /* fin de else if */
    }
}
```

```

else { /* ignora el valor duplicado del dato */
printf( "dup" );
} /* fin de else */

} /* fin de else */

} /* fin de la función insertaNodo */

/* comienza el recorrido inorden del árbol */
void inOrden( ptrNodoArbol ptrArbol )
{

/* si el árbol no está vacío, entonces recórrelo */
if ( ptrArbol != NULL ) {
inOrden( ptrArbol->ptrIzq );
printf( "%3d", ptrArbol->dato );
inOrden( ptrArbol->ptrDer );
} /* fin de if */

} /* fin de la función inOrden */

/* comienza el recorrido preorden del árbol */
void preOrden( ptrNodoArbol ptrArbol )
{

/* si el árbol no está vacío, entonces recórrelo */
if ( ptrArbol != NULL ) {
printf( "%3d", ptrArbol->dato );
preOrden( ptrArbol->ptrIzq );
preOrden( ptrArbol->ptrDer );
} /* fin de if */

} /* fin de la función preOrden */

/* comienza el recorrido postOrden del árbol */
void postOrden( ptrNodoArbol ptrArbol )
{

/* si el árbol no está vacío, entonces recórrelo */
if ( ptrArbol != NULL ) {
postOrden( ptrArbol->ptrIzq );
postOrden( ptrArbol->ptrDer );
printf( "%3d", ptrArbol->dato );
} /* fin de if */

} /* fin de la función postOrden */

```

Fuente: C/ C++ y Java Cómo Programar

@telecristy.

f COMPARTIR EN FACEBOOK

t COMPARTIR EN TWITTER

g+ COMPARTIR EN GOOGLE+

¿Te gustó esta publicación? Márcala como favorita ★



@telecristy

@telecristy

Estados Unidos

<https://www.codejobs.biz/>

Publicaciones del autor

Favoritos del autor

ARTÍCULOS RELACIONADOS



El lenguaje de programación GO



Concurrent Versions System (CVS)



37 Javascript - Comprobando igualda...



Preguntas de una entrevista de PHP ...



¡Feliz día del Programador!



Conectar a una base de datos MySQL ...



0 Comments

Codejobs

1 Login

Recomendar

Compartir

ordenar por el mejor

Comienza la discusión...

Sé el primero en comentar.

TAMBIÉN EN CODEJOBS

Recursion en JavaScript - Aprende a Programar

2 comments • hace 7 meses•

Nico — Lo que pasa con la recursión es que va dejando tareas pendientes en memoria, en el ejemplo que dieron power(2,3), el resultado parcial sería base*power(2,2),

Desarrollando un CMS desde cero con Node.js Express, Handlebars, Stylus, Gulp y más: Gulp #3 -

1 comment • hace un año•

Jose Gelimer Gomez — Porque no pones TODO el codigo de eslint???

Argumentos opcionales en funciones - Javascript - Aprende a Programar

1 comment • hace 8 meses•

esutoraiki — Esta es una manera, pero en la actualidad (2016) ya existen mejores maneras, por ejemplo con el uso de ES6: function power(base, exponent = 2) { return

JavaScript es fácil: ¿Qué es el Event Delegation? #2 - Aprende a Programar

1 comment • hace un año•

Eruz Apodaca — mindfuck del minuto 0:30 al 0:50

Subscribe

Agrega Disqus a tu sitioAdd DisqusAdd

Privacidad

DISQUS

Tecnologías

Codejobs es una comunidad de amantes a la tecnología que creemos que el conocimiento es la única forma de ser verdaderamente libres y autónomos.

© 2017 Codejobs

https://www.codejobs.biz/es/blog/2013/10/30/creacion-y-recorrido-de-un-arbol-binario-en-c

4/4